PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant	:	Randy Keith Lomnes
Application No.	:	09/923,727
Filed	:	August 6, 2001
For	:	METHOD AND SYSTEM FOR AUTOMATICALLY PRESERVING PERSISTENT STORAGE
		Examiner : Shen Jen Tsai
		Art Unit : 2186
		Docket No. : 470039.401

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

DECLARATION OF RANDY KEITH LOMNES, Ph.D.
PURSUANT TO 37 CFR 1.132

I, Randy Keith Lomnes, residing at 7934 Swanson View Drive, Pender Island, B.C. Canada V0N2M2, declare as follows:

1. I am the inventor of the invention claimed in the above-identified patent application.

2. I am a co-founder and Director of Hyper Technologies, Inc.

3. I have a Ph.D., M.S., and B.S. in Physics, obtained from University of Alberta, Edmonton, Canada. Since obtaining my Ph.D., I have founded or have been part of the ownership of 5-10 companies that develop products for, or perform services related to, hardware and/or software design. I have extensive experience in both hardware (*e.g.*, computer system peripheral card design) and software design/implementation. I am also a named inventor on three issued, electronics-related, U.S. patents.

4. I have studied U.S. Patent No. 6,092,161 to White et. al (hereinafter "White") in detail. White describes a peripheral (auxiliary hardware) card that is situated between a personal computer system (PC) and a disk drive. The peripheral card of White has a separate, supervisor computer processor that executes firmware instructions (*i.e.*, instructions obtained from a ROM such as ROM 326 of Figure 8) situated on the peripheral card, in order to control reading from and writing to a disk, thereby protecting the disk from contamination with malicious software, such as viruses.

5. The peripheral card of White is not the same thing as a software redirection driver, which is implemented as part of an operating system. I know this because the peripheral card of White physically sits between the personal computer and the disk drive, and is connected to each by ribbon cables. For example, Figure 6 shows a ribbon cable 201 that is used to connect the peripheral card to the personal computer, and a ribbon cable 202 that is used to connect the card to the disk drive. The text on lines 35 through 42 of column 9 explains that the ribbon cable 201 connects to the personal computer, and that the ribbon cable 202 connects the card to the disk drive. It also says that all communication between the personal computer and the disk is controlled by the card. The peripheral card of White is completely separate from, and does not interact with, the operating system of the personal computer.

6. By reading White and knowing how such cards work, I understand that the peripheral card of White intercepts read and write accesses that are generated by the operating system and that are directed to particular disk partitions, and then to redirect those read and write accesses to other disk partitions. The peripheral card of White does not appear to have any understanding of the file system managed by the operating system.

7. White only describes a hardware/firmware-based approach, as can be seen in Figures 6-8, and the accompanying text. Nowhere does White describe using a software driver-based approach.

8. White does not consider a software-based approach. The purpose stated by White and the design of the separate peripheral card is to protect the disk from malicious software.

Having a separate card that controls all accesses to the disk can do this. In contrast, software driver-based approaches are susceptible to tampering. Therefore, modifying the hardware-oriented approach of White to operate as a software driver does not make any sense, because hackers could tamper with disk accesses. Therefore, a software driver would defeat the purpose of protecting the disk from tampering.

9. The peripheral card of White does not operate by executing instructions from a volatile memory. As is clear from Figures 6-8, and the accompanying text, the peripheral card of White executes instructions from a ROM. If instead White executed instructions from a volatile memory, such as a RAM, the RAM would have to first (upon initialization or power up) be loaded with instructions from somewhere, such as from a hard disk or other storage device. However, during the loading of instructions into RAM, and before the peripheral card can function to protect the hard disk, there would be a time period during which the hard disk could not be protected from malicious accesses. Thus, it does not make any sense to have White execute instructions from RAM or other volatile memory, because it would defeat White's purpose.

10. White also does not discuss any way to load instructions in the ROM once the card is in use. I know from my experience that the types of ROM used for instruction storage by the peripheral card of White are typically burned by a manufacturer. Such a ROM is ordinarily upgraded by replacing the ROM chip itself. In addition, Figures 6-8 do not appear to show any of the circuits that would be needed to program the ROM from the card itself. For example, chip 326 of Figure 8 does not appear to have any programming lines associated with it. Furthermore, to keep the peripheral card as tamper proof as possible, one would not want to allow programming, because then malicious software could possibly modify the operation of the peripheral card, thereby disabling disk protection.

I hereby declare that all statements made herein are, to my own knowledge, true and that all statements made on information or belief are believed to be true; and further that these statements are made with the knowledge that willful false statements and the like are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the captioned patent application or any patent issued therefrom.

Date June 13, 2007

Randy Keith Lomnes, Ph.D.

972155_1.DOC